

3/3/1 (Item 1 from file: 351)  
DIALOG(R)File 351:Derwent WPI  
(c) 2006 Thomson Derwent. All rts. reserv.

015281829 \*\*Image available\*\*  
WPI Acc No: 2003-342761/200332  
XRPX Acc No: N03-274152

**Parallel calculation method has a number of processors**

Patent Assignee: JAPAN SCI & TECHNOLOGY CORP (NISC-N); NEC CORP (NIDE );  
KAGAKU GIJUTSU SHINKO JIGYODAN (KAGA-N); NEC JOHO SYSTEMS KK (NIDE );  
NIPPON DENKI SOFTWARE KK (NIDE ); MURASE T (MURA-I); NAKATA K (NAKA-I);  
SAKUMA T (SAKU-I); TAKADA T (TAKA-I)

Inventor: MURASE T; NAKATA K; SAKUMA T; TAKADA T

Number of Countries: 007 Number of Patents: 005

**Patent Family:**

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200327873	A1	20030403	WO 2002JP9871	A	20020925	200332 B
JP 2003099408	A	20030404	JP 2001292092	A	20010925	200332
EP 1443415	A1	20040804	EP 2002768061	A	20020925	200451
			WO 2002JP9871	A	20020925	
KR 2004054693	A	20040625	KR 2004704222	A	20040323	200470
US 20040260529	A1	20041223	WO 2002JP9871	A	20020925	200504
			US 2004490703	A	20040817	

Priority Applications (No Type Date): JP 2001292092 A 20010925

**Patent Details:**

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200327873	A1	J	10	G06F-015/16	
				Designated States (National):	CN KR US
				Designated States (Regional):	DE FR GB
JP 2003099408	A		9	G06F-015/16	
EP 1443415	A1	E		G06F-015/16	Based on patent WO 200327873
				Designated States (Regional):	DE FR GB
KR 2004054693	A			G06F-015/16	
US 20040260529	A1			G06F-013/12	

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-099408

(43)Date of publication of application : 04.04.2003

(51)Int.Cl.

G06F 15/16  
C07B 61/00  
G06F 15/177  
G06F 17/13  
G06F 19/00

(21)Application number : 2001-292092

(71)Applicant : JAPAN SCIENCE & TECHNOLOGY  
CORP  
NEC CORP  
NEC SOFT LTD  
NEC INFORMATEC SYSTEMS LTD

(22)Date of filing : 25.09.2001

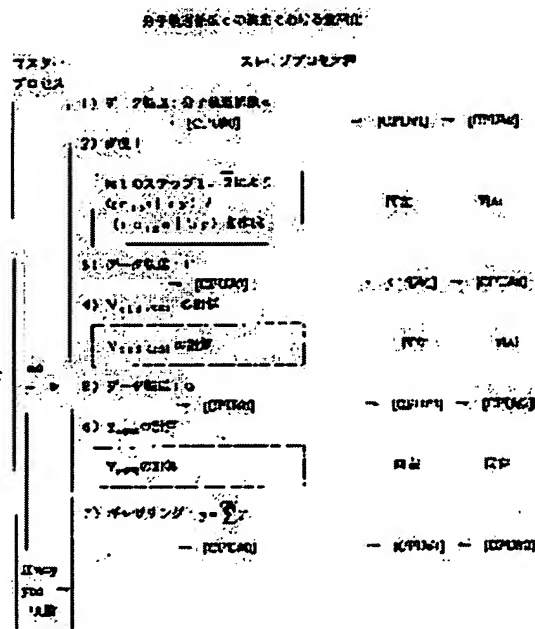
(72)Inventor : NAKADA KAZUTO  
MURASE TADASHI  
SAKUMA TOSHIHIRO  
TAKADA TOSHIKAZU

## (54) PARALLEL CALCULATION METHOD

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To cancel the increase in costs for calculation accompanying size growing of calculation objective molecules in parallel processing.

**SOLUTION:** In a calculation method using a parallel computer having a plurality of processors, when transforming 2 electronic integration from the atomic orbital base (rs&verbar;tu) to a molecular orbital base (ab&verbar;cd), the indexes (r) and (s) of the atomic orbital to be calculated are designated to the plurality of processors to obtain a parallel calculation method for performing processing concerning all the combinations of designated indexes R and S and the indexes (t) and (u) of the atomic orbital to be calculated in each of the plurality of processors. Thus, grain sizes are equalized to connect a large number of commodity processors under high performance, thereby costs for a high speed arithmetic computer are reduced. In addition, since an inexpensive local memory can be utilized, a large main storage area can be secured as the whole computer.



## LEGAL STATUS

[Date of request for examination]

05.08.2005

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or

application converted registration]  
[Date of final disposal for application]  
[Patent number]  
[Date of registration]  
[Number of appeal against examiner's decision  
of rejection]  
[Date of requesting appeal against examiner's  
decision of rejection]  
[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-99408 ✓

(P2003-99408A)

(43) 公開日 平成15年4月4日 (2003.4.4)

(51) Int.Cl. <sup>7</sup>	識別記号	FI	テーマコード(参考)
G 0 6 F 15/16	6 1 0	G 0 6 F 15/16	6 1 0 Z 4 H 0 0 6
C 0 7 B 61/00		C 0 7 B 61/00	Z 5 B 0 4 5
G 0 6 F 15/177	6 7 4	G 0 6 F 15/177	6 7 4 A 5 B 0 5 6
17/13		17/13	
19/00	1 1 0	19/00	1 1 0

審査請求 未請求 請求項の数6 OL (全 9 頁)

(21) 出願番号 特願2001-292092(P2001-292092)

(22) 出願日 平成13年9月25日 (2001.9.25)

(71) 出願人 396020800

科学技術振興事業団

埼玉県川口市本町4丁目1番8号

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(71) 出願人 000232092

NECソフト株式会社

東京都江東区新木場一丁目18番6号

(74) 代理人 100071272

弁理士 後藤 洋介 (外1名)

最終頁に続く

(54) 【発明の名称】 並列計算方法

(57) 【要約】

【課題】 並列処理で計算対象分子の大型化に伴う計算コストの増大を解消する。

【解決手段】 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、2電子積分の原子軌道基底 ( $rs|tu$ ) から分子軌道基底 ( $ab|cd$ ) への変換の際に、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス  $r$  と  $s$  を指定し、前記複数のプロセッサの各々において、指定されたインデックス  $R$  と  $S$  と、計算すべき原子軌道のインデックス  $t$  と  $u$  の全ての組合せとについての処理を行う並列計算方法が得られ、これにより、粒度が均一になり、高性能下でコモディティプロセッサを多数接続することができ、高速演算コンピュータの費用低減が可能になり、更に、廉価なロカルメモリを活用できるので、コンピュータ全体として大きな主記憶領域を確保できる。

分子軌道基底  $c$  の決定における並列化マスター  
プロセス

スレーブプロセス群

- 1) データ転送: 分子軌道基底  $c$   
→ [CPU#0] → [CPU#1] → [CPU#2]
- 2) 変換 1  
図1のステップ1、2により  
( $rc_{rs}|xy$ ) と  
( $rc_{rs}|by$ ) を作成  
同左 同左
- 3) データ転送:  $\Gamma$   
→ [CPU#0] → [CPU#1] → [CPU#2]
- 4)  $V_{cbs}(rs)$  の計算  
 $V_{cbs}(rs)$  の計算  
同左 同左
- 5) データ転送:  $u$   
→ [CPU#0] → [CPU#1] → [CPU#2]
- 6)  $Y_{cbs}$  の計算  
 $Y_{cbs}$  の計算  
同左 同左
- 7) ギャザリング:  $y = \sum Y$   
← [CPU#0] ← [CPU#1] ← [CPU#2]

no

if any  
yes →  
収束

## 【特許請求の範囲】

【請求項1】 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、2電子積分の原子軌道基底 (rs | tu) から分子軌道基底 (ab | cd) への変換の際に、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス r と s を指定し、前記複数のプロセッサの各々において、指定されたインデックス R と S と、計算すべき原子軌道のインデックス t と u の全ての組合せとについての処理を行うことを特徴とする並列計算方法。

【請求項2】 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、2電子積分の原子軌道基底 (rs | tu) から分子軌道基底 (ab | cd) への変換の際に、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス r と s を指定し、前記複数のプロセッサの各々において、指定されたインデックス r と s と、計算すべき原子軌道のインデックス t と u の全ての組合せとについての処理を行うことにより、前記複数のプロセッサにおける仕事量を表わす粒度が均一になるように処理すべきデータを分割することを特徴とする並列計算方法。

【請求項3】 請求項1又は2に記載の並列計算方法において、前記2電子積分の計算途中では前記複数のプロセッサ間ではデータ転送を行わず、前記複数のプロセッサの前記処理の後に一度だけギャザリングして前記複数のプロセッサの処理されたデータの和を取ることを特徴とする並列計算方法。

【請求項4】 請求項1又は2に記載の並列計算方法において、前記複数のプロセッサの各々は、

- 1) プロセッサ毎に指定された R、S と t、u の全ての組合せとについて原子軌道基底での2電子積分 (RS | tu) を計算し、
- 2) 計算された2電子積分について、u から d への変換を行い、
- 3) 上記2) で変換されたデータについて、t から c への変換を行い、
- 4) 上記3) で変換されたデータについて、S から b の構成要素である B への変換を行い、
- 5) 上記4) で変換されたデータについて、R から a の構成要素である A への変換を行い、
- 6) 上記5) で変換されたデータを各プロセッサのローカルメモリーに保存し、
- 7) 最後にギャザリングして分子軌道基底での2電子積分 (ab | cd) を求めることを特徴とする並列計算方法。

【請求項5】 請求項1又は2に記載の並列計算方法において、前記複数のプロセッサの各々は、

- 1) プロセッサ毎に指定された R、S と t、u の全ての

組合せとについて原子軌道基底での2電子積分 (RS | tu) を計算し、

- 2) 計算された2電子積分について、u から d への変換を行い、

- 3) 上記2) で変換されたデータについて、t から c への変換を行い、

- 4) 上記3) で変換されたデータについて、S から b の構成要素である B への変換を行い、

- 5) R を変換せず  $V_{c b d}$  (RS) として前記複数のプロセッサの各々のローカルメモリーに保存し、

- 6) 分子軌道の係数の決定に必要な量である  $y_{a c}$  の繰り返し計算において、R から a への変換係数と分子軌道間の変換係数と前記  $V_{c b d}$  (RS) との積和を計算し、計算結果を前記複数のプロセッサの各々のローカルメモリーに保存することを特徴とする並列計算方法。

【請求項6】 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、分子軌道基底での2次の密度行列  $\Gamma$  を前記複数のプロセッサの各々に転送し、前記複数のプロセッサの各々において、分子軌道基底から原子軌道基底に逆変換することにより、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス r と s を指定し、前記複数のプロセッサの各々において、指定されたインデックス R と S と、計算すべき原子軌道のインデックス t と u の全ての組合せとについての処理を行い、2電子積分の原子核座標による微分計算において、前記複数のプロセッサにおける仕事量を表わす粒度が均一になるように処理すべきデータを分割することを特徴とする並列計算方法。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、非経験的分子軌道法の内、多配置自己無撞着場 (Multi Configuration Self Consistent Field: MCSCF) 法及び配置間相互作用 (Configuration Interaction: CI) 法による全エネルギーとエネルギー勾配の並列計算手法に関する。

【0002】

【従来の技術】非経験的分子軌道計算法において、電子相関を取り入れることのできる代表的な計算手法として、上記 MCSCF 法と CI 法が確立されている。MCSCF 法では、全エネルギー E と原子核の座標による微分即ちエネルギー勾配 (原子核に働く力) とは、

【数1】

$$E = \sum_{ab} \gamma_{ab} h_{ab} + \frac{1}{2} \sum_{abcd} \Gamma_{abcd} (ab | cd) \quad (1)$$

【数2】

$$\frac{\partial E}{\partial q} = \sum_{ab} \gamma_{ab} \frac{\partial h_{ab}}{\partial q} + \frac{1}{2} \sum_{abcd} \Gamma_{abcd} \frac{\partial (ab | cd)}{\partial q} \quad (2)$$

のように与えられる。ここで、MO は分子軌道 (Molecular Orbital)、q は分子を構成する原子の原子核の座標 x、y、z のいずれかである。γ 及び Γ は後述の解法により

求められる電子配置の係数Cの関数であり、 $h_{ab}$ と $(ab|cd)$ は分子軌道基底による1電子積分及び2電子積分で、原子軌道基底の $h_{rs}$ と $(rs|tu)$ とから得られる。

$$h_r = \iiint x_r(r_1) h x_r(r_1) dr_1, \quad h = -\frac{\hbar^2}{2} \nabla^2 - \sum_{i=1}^{Nuc} \frac{Z_i}{|r_1 - R_i|} \quad (3)$$

$$(r_2 | r_1, \quad h = h/2\pi)$$

【数4】

$$(rs|tu) = \iiint \iiint x_r(r_1) x_s(r_1) \frac{1}{|r_1 - r_2|} x_t(r_2) x_u(r_2) dr_1 dr_2 \quad (4)$$

のように定義される。ここで、 $h$ はプランク定数、NucはNucleiの略で原子の数を表わす。 $Z_i$ は原子核の電荷、 $R_i$ は原子核の位置を表わす。これら式(3)及び式(4)の積分は、分子軌道 $\phi_a$ と原子軌道 $x_r$ の

【数5】

$$\phi_a = \sum_r c_{ra} x_r \quad (5)$$

なる関係から、

【数6】

$$h_{ab} = \sum_r \sum_s c_{ra} c_{sb} h_{rs} \quad (6)$$

【数7】

$$(ab|cd) = \sum_r \sum_s \sum_t \sum_u c_{ra} c_{sb} c_{tc} c_{ud} (rs|tu) \quad (7)$$

のように原子軌道基底から分子軌道基底へ変換される。ここで、 $N$ は原子軌道の数で、本発明では1000以上を想定しているが、それ以下の場合についても本手法を適用することは問題なく可能である。 $c_{ra}$ は式(5)で示される、原子軌道を分子軌道の変換する変換行列

$$y_{ac} = \sum_b \sum_d \sum_x \sum_y \{ (ab|xy) \Gamma_{cxy} + 2(ax|by) \Gamma_{cxy} \} u_{bd} \quad (10)$$

なる量が必要になる。ここで、 $u_{bd}$ は分子軌道の線形変換に関わる行列である。

【0005】MCSCF法、CI法のいずれにおいても、分子軌道基底の2電子積分 $(ab|cd)$ の生成が計算コストの大半を占めている。MCSCF法の中で現在主流となっているCASSCF(Complete Active Space SCF)では、電子励起を限られた分子軌道の範囲に限定することで、定式の簡素化を図っている。そのActive Space空間に属する分子軌道の数 $n$ 、原子軌道基底の数 $N$ とすると、一般には $n \ll N$ なる関係が成立する。従来の計算スキームでは、原子軌道基底の2電子積分を全て主記憶もしくはディスクなどの外部記憶媒体に保存し、式(7)の変換を行っている。その変換アルゴリズムを、図5及び図6に示す。分子軌道のインデックス $a, b, c, d$ と原子軌道のインデックス $r, s, t, u$ からなる単純な8重のDOループでは $n^4 N^4$ 回の掛け算が必要となるが、このアルゴリズムでは5重のDOループを4回繰り返すことで同等の結果が得られ、その演

$h_{rs}$ と $(rs|tu)$ は、

【数3】

で、分子軌道の係数と呼ばれる。 $c_{sb}$ も同様の分子軌道の係数である。MCSCF法では、電子配置の係数Cと分子軌道の係数 $c$ の両方を変分法により決めるが、CI法ではCのみを求める所が異なっている。

【0003】電子配置の係数Cは、次の方程式から求められる。即ち、

【数8】

$$\sum_j (H_{ij} - \delta_{ij} E) C_j = 0 \quad (8)$$

【数9】

$$H_{ij} = \sum_{ab} y_{ab}^i h_{ab} + \frac{1}{2} \sum_{abcd} \Gamma_{abcd}^i (ab|cd) \quad (9)$$

ここで、CSFはconfiguration state function(電子配置関数)で、波動関数がこの反対称化された行列式CSFの線形結合で与えられる。 $\delta_{ij}$ はクロネッカのデルタで、 $i=j$ なら1で、それ以外は0である。

【0004】また、分子軌道の係数の決定には、

【数10】

算数は $nN^4 + n^2N^3 + n^3N^2 + n^4N$ となる。例えば、 $n=10$ 、 $N=1000$ とすると、約1000倍の高速化が図れることになる。しかしながら、この方法では、原子軌道基底の2電子積分や変換途中の中間データを保存するためのコンピュータリソースが膨大になり、大型分子が計算できないという本質的な問題がある。

【0006】

【発明が解決しようとする課題】近年のコンピュータはパラレルプロセッサの搭載により高速性を実現しており、MCSCF法及びCI法においても、並列処理により計算対象分子の大型化と計算コストの削減を実現する必要がある。パラレルコンピュータの利点は、

①コモディティプロセッサを多数接続することにより、高速演算コンピュータを廉価に実現できる。

【0007】②多数のプロセッサのロカールメモリを活用できるので、コンピュータ全体として大きな主記憶領域を確保できる。

【0008】である。

【0009】原子軌道基底及び分子軌道基底での2電子積分は互いに独立であり、この特徴を活用して並列化しようとする、次の問題が生じる。即ち、分子軌道基底での2電子積分をひとつ計算するのに、 $N^4$ 個の原子軌道基底の2電子積分全てが必要になるので、

1) 原子軌道基底での2電子積分の一部をプロセッサに分担させる分散処理をすると、並列化により積分計算時間は短縮できるが、 $N^4$ 個の2電子積分全てを各々のプロセッサに集めなければならず、全プロセッサ間で相互通信が発生し通信ネックとなる。

【0010】2) 通信ネックを避けようとするれば、 $N^4$ 個の原子軌道基底の積分を全て各々のプロセッサで計算しなければならず、並列化による計算時間の短縮にはつながらない。

【0011】という相反する問題が生じる。

【0012】2電子積分の原子核座標による微分についても、式(2)の計算に分子軌道基底の積分が必要であり、全く同様の議論が成立する。

【0013】本発明の目的は、上述した問題を解決することができる、複数のプロセッサを有するパラレルコンピュータを用いた並列計算方法を提供することにある。

【0014】

【課題を解決するための手段】本発明による並列計算方法は、以下の通りである。

【0015】(1) 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、2電子積分の原子軌道基底( $r s | t u$ )から分子軌道基底( $a b | c d$ )への変換の際に、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス $r$ と $s$ を指定し、前記複数のプロセッサの各々において、指定されたインデックス $R$ と $S$ と、計算すべき原子軌道のインデックス $t$ と $u$ の全ての組合せとについての処理を行うことを特徴とする並列計算方法。

【0016】(2) 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、2電子積分の原子軌道基底( $r s | t u$ )から分子軌道基底( $a b | c d$ )への変換の際に、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス $r$ と $s$ を指定し、前記複数のプロセッサの各々において、指定されたインデックス $r$ と $s$ と、計算すべき原子軌道のインデックス $t$ と $u$ の全ての組合せとについての処理を行うことにより、前記複数のプロセッサにおける仕事量を表わす粒度が均一になるように処理すべきデータを分割することを特徴とする並列計算方法。

【0017】(3) 上記(1)又は(2)に記載の並列計算方法において、前記2電子積分の計算途中では前記複数のプロセッサ間ではデータ転送を行わず、前記複数のプロセッサの前記処理の後に一度だけギャザリングして前記複数のプロセッサの処理されたデータの和を取

ることを特徴とする並列計算方法。

【0018】(4) 上記(1)又は(2)に記載の並列計算方法において、前記複数のプロセッサの各々は、

1) プロセッサ毎に指定された $R$ 、 $S$ と $t$ 、 $u$ の全ての組合せとについて原子軌道基底での2電子積分( $R S | t u$ )を計算し、

2) 計算された2電子積分について、 $u$ から $d$ への変換を行い、

3) 上記2)で変換されたデータについて、 $t$ から $c$ への変換を行い、

4) 上記3)で変換されたデータについて、 $S$ から $b$ の構成要素である $B$ への変換を行い、

5) 上記4)で変換されたデータについて、 $R$ から $a$ の構成要素である $A$ への変換を行い、

6) 上記5)で変換されたデータを各プロセッサのローカルメモリーに保存し、

7) 最後にギャザリングして分子軌道基底での2電子積分( $a b | c d$ )を求めることを特徴とする並列計算方法。

【0019】(5) 上記(1)又は(2)に記載の並列計算方法において、前記複数のプロセッサの各々は、

1) プロセッサ毎に指定された $R$ 、 $S$ と $t$ 、 $u$ の全ての組合せとについて原子軌道基底での2電子積分( $R S | t u$ )を計算し、

2) 計算された2電子積分について、 $u$ から $d$ への変換を行い、

3) 上記2)で変換されたデータについて、 $t$ から $c$ への変換を行い、

4) 上記3)で変換されたデータについて、 $S$ から $b$ の構成要素である $B$ への変換を行い、

5)  $R$ を変換せず $V_{c b d}(R S)$ として前記複数のプロセッサの各々のローカルメモリーに保存し、

6) 分子軌道の係数の決定に必要な量である $y_{a c}$ の繰り返し計算において、 $R$ から $a$ への変換係数と分子軌道間の変換係数と前記 $V_{c b d}(R S)$ との積和を計算し、計算結果を前記複数のプロセッサの各々のローカルメモリーに保存することを特徴とする並列計算方法。

【0020】(6) 複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、分子軌道基底での2次の密度行列 $\Gamma$ を前記複数のプロセッサの各々に転送し、前記複数のプロセッサの各々において、分子軌道基底から原子軌道基底に逆変換することにより、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス $r$ と $s$ を指定し、前記複数のプロセッサの各々において、指定されたインデックス $R$ と $S$ と、計算すべき原子軌道のインデックス $t$ と $u$ の全ての組合せとについての処理を行い、2電子積分の原子核座標による微分計算において、前記複数のプロセッサにおける仕事量を表わす粒度が均一になるように処理すべきデータを分割することを特徴とする並列計算方法。

【0021】

【発明の実施の形態】次に本発明の実施の形態について説明する。

【0022】本願発明者は、原子軌道基底での2電子積分及び原子核座標による微分いずれも相互に独立であるので、原子軌道基底から分子軌道基底への変換において、原子軌道のインデックスrとsで並列化し、(すなわち、前記複数のプロセッサに対して、計算すべき原子軌道のインデックスrとsを指定し、tとuの全ての組み合わせを各プロセッサで処理することで、粒度が均一になるように処理データを分割することを考える。インデックスrとsで並列化するが、t及びuの全ての組み合わせについて各プロセッサで計算するので、2電子積分1個当たり平均1万回の浮動小数点演算が必要であることから、各プロセッサ当たり10000N<sup>2</sup>の演算を行うことになり、充分大きな粒度を確保できる。以下で詳しく述べるが、並列性能の向上には、粒度の均一化と最大化及びプロセッサ間の転送回数の最小化が必要条件である。本計算手順では、2電子積分計算途中ではプロセッサ間でデータ転送を行わず、変換後にギャザリングを1度行うことを特徴としている。

【0023】電子配置の係数Cを求める解法においては、原子軌道のインデックスr、sで並列化を行い、各プロセッサでの計算処理を、

1) プロセッサ毎に指定されたR、Sに対してt、uの全てについて原子軌道基底での2電子積分(RS|tu)を計算する。

【0024】2) 計算された2電子積分について、uからdへの変換を行う。

【0025】3) tからcへの変換を行う。

【0026】4) SからBへの変換を行う。

【0027】5) RからAへの変換を行う。

【0028】の手順で行い、各プロセッサのローカルメモリにこの変換データを保存し、最後にギャザリングして分子軌道基底での2電子積分(ab|cd)を求める。

【0029】また、分子軌道の係数cを求める繰り返し解法においては、原子軌道のインデックスr、sで並列化を行い、各プロセッサでの計算処理を、

1) プロセッサ毎に指定されたR、Sに対してt、uの全てについて原子軌道基底での2電子積分(RS|tu)を計算する。

$$\frac{\partial E}{\partial q} = \sum_{rs} AO \gamma_r \frac{\partial h_r}{\partial q} + \sum_{rstu} \Gamma_{rstu} \frac{\partial (rs|tu)}{\partial q} - \sum_{rs} AO w_r \frac{\partial S_r}{\partial q} \quad (11)$$

に基づいて並列計算する。

【0041】ここで、AOは原子軌道(Atomic Orbital)である。Wはエネルギー勾配法に現れる量で、軌道エネルギーと分子軌道の係数の積で与えられる。また、S<sub>r</sub>sは重なり積分である。

【0030】2) 計算された2電子積分について、uからdへの変換を行う。

【0031】3) tからcへの変換を行う。

【0032】4) SからBへの変換を行う。

【0033】5) Rを変換せずV<sub>c b d</sub>(RS)としてローカルメモリに保存する。

【0034】6) y<sub>a c</sub>の繰り返し計算において、Rからaへの変換係数(c<sub>R a</sub>)と分子軌道間の変換係数(u<sub>b d</sub>)との積和を同時に計算する(後述する式(13)参照)。

【0035】の手順で行い、各プロセッサのローカルメモリにデータ量の少ないV<sub>c b d</sub>(RS)を保存することで、繰り返し計算における主記憶上の作業領域の増大を防いでいる。

【0036】エネルギー勾配計算において、分子軌道基底での2次の密度行列Γを各プロセッサに転送し、分子軌道基底から原子軌道基底に逆変換することにより、原子軌道のインデックスr、sで並列化することができ、2電子積分の場合と同様の議論が成立する。この場合、原子軌道rについてのみ、原子核の座標での微分が必要となることに注意する。

【0037】次に本発明の実施例について説明する。

【0038】MCSCF法における電子配置の係数Cと分子軌道の係数c、CI法における電子配置の係数Cを求める解法においては、主記憶上の作業領域を削減するため、繰り返し解法が広く使われている。そのため、分子軌道基底の2電子積分などの中間データを保存する必要があるが、並列コンピュータでは上述した利点②(多数のプロセッサのローカルメモリを活用できるので、コンピュータ全体として大きな主記憶領域を確保できる。)により可能である。一方、エネルギー勾配計算では繰り返し計算の必要性がないので、本発明では、次の1)及び2)なる基本的な考え方に立脚して、並列化アルゴリズムを開発する。

【0039】1) 電子配置の係数C及び分子軌道の係数cの決定においては、n<Nの関係からデータ量の少ない分子軌道基底での2電子積分などの中間データを、各プロセッサの主記憶上に保存し繰り返し使用する。

【0040】2) エネルギー勾配計算においては、分子軌道基底のγ及びΓを原子軌道基底に逆変換して、原子軌道基底での表式

【数11】

$$\frac{\partial E}{\partial q} = \sum_{rs} AO \gamma_r \frac{\partial h_r}{\partial q} + \sum_{rstu} \Gamma_{rstu} \frac{\partial (rs|tu)}{\partial q} - \sum_{rs} AO w_r \frac{\partial S_r}{\partial q} \quad (11)$$

【0042】まず最初に、電子配置の係数Cの決定に必要なActive Spaceに属する分子軌道基底での2電子積分の生成法について述べる。原子軌道のインデックスr、sで並列化を行い、t、uの全てについて各々のプロセッサで原子軌道基底の2電子積分(RS|tu)を計算



する。その基本アルゴリズムを図 1 及び図 2 に示す。ここで、 $r$ 、 $s$  で指定される原子軌道のインデックスの内、 $R$ 、 $S$  は或るプロセッサに割り当てられた原子軌道

$$(ab|cd) = \sum_A \sum_B (c_{RA} R c_{SB} S | cd) = \sum_A \sum_B (AB|cd) \quad (12)$$

と書き直せる。ここで、 $A$ 、 $B$  は、式 (5) の関係から分子軌道  $a$ 、 $b$  に対する原子軌道  $r$ 、 $s$  の線形結合成分の内、各プロセッサに対して指定された  $R$ 、 $S$  のみからなっているので、最後に  $(AB|cd)$  をギャザリングして初めて正しい  $(ab|cd)$  になることに注意する。この関係から、各プロセッサでの計算手順は、

1) プロセッサ毎に指定された  $R$ 、 $S$  に対して  $t$ 、 $u$  の全てについて原子軌道基底での 2 電子積分  $(RS|tu)$  を計算する。

【0043】 2) 計算された 2 電子積分について、 $u$  から  $d$  への変換を行う。

【0044】 3)  $t$  から  $c$  への変換を行う。

【0045】 4)  $S$  から  $B$  への変換を行う。

【0046】 5)  $R$  から  $A$  への変換を行う。

【0047】 となる。ここで注意すべき点は、1) ~ 5) のステップの実行においてプロセッサ間の通信は全く発生しておらず、且つ計算すべき原子軌道基底での 2 電子積分は全プロセッサを合わせて  $N^4$  個であることで

$$\begin{aligned} y_{ac} &= \sum_R \sum_S \sum_t \sum_y \{ (ab|xy) \Gamma_{cdy} + 2(ax|by) \Gamma_{cdy} \} \mu_{bd} \\ &= \sum_R \sum_S \sum_t \sum_y \left\{ \sum_A \sum_B (c_{RA} R c_{SB} S | xy) \Gamma_{cdy} + 2 \sum_A \sum_B (c_{RA} R c_{SB} S | by) \Gamma_{cdy} \right\} \mu_{bd} \\ &= \sum_R \sum_S \sum_t \sum_y \sum_A \sum_B \{ (c_{RA} R c_{SB} S | xy) \Gamma_{cdy} + 2(c_{RA} R c_{SB} S | by) \Gamma_{cdy} \} \mu_{bd} \\ &= \sum_R \sum_S c_{RS} \sum_t \sum_y \sum_A \sum_B \{ (R c_{SB} S | xy) \Gamma_{cdy} + 2(R c_{SB} S | by) \Gamma_{cdy} \} \mu_{bd} \\ &= \sum_R \sum_S c_{RS} \sum_t \sum_y V_{cd(RS)} \mu_{bd} \\ &= \sum_R \sum_S Y_{ac(RS)} \quad (13) \end{aligned}$$

のようになる。

【0049】 更に、 $Y_{ac}(RS)$  を各プロセッサ(CPU: Central Processing Unit)で計算した部分とすると、式 (12) からの推察により、同様にギャザリングすることができるがわかる。その時の転送データ量は、 $N^2$  であることに注意する。従って、 $r$  と  $s$  で並列化し、各プロセッサで計算された  $Y_{ac}(RS)$  をギャザリングすれば  $y_{ac}$  が求められる。そのデータの流れを、図 3 に示す。ここで、 $V_{cbd}(RS)$  として各プロセッサに保存するのは、以下の理由による。式 (10) の  $a$ 、 $b$  は原則全ての分子軌道になるので次元数は  $N$  であるのに対して、 $x$ 、 $y$  は  $n$  である。よって、その大きさは第 1、2 項とも  $n^2 N^2$  となる。 $r$  の変換を行わず保存することで、演算数の増加をきたすが、作業領域を  $n^2 N$  に抑えることができる。また、 $(ab|xy)$  をギャザリングすると、そのデータ量は  $n^2 N^2$  であるが、 $x$ 、 $y$  に

である。今、

【数 12】

ある。最後に、 $(ab|cd)$  を求めるため、各プロセッサ上の  $(AB|cd)$  をギャザリングすることになるが、その転送量は高々  $n^4$  である。 $n$  は 10 のオーダーであり、且つマスタープロセッサへの一方向転送であるので、転送時間が問題になることはない。また、プロセッサ台数が少ない場合には、 $R$  と  $S$  を適当にグループ化すれば、同様に並列化できることは明らかである。全体の演算数については、従来方式が前述のように  $n N^4 + n^2 N^3 + n^3 N^2 + n^4 N$  であるが、本方式ではステップ 1 ~ 4 の演算が  $N^2$  個のプロセッサで発生するので、 $N^2 (n N^2 + n^2 N + n^3 + n^4) + n^4 N^2$  となり、 $n^4 N (N-1) + n^4 N^2$  だけ並列化により演算数が増えていることになる。

【0048】 次に、分子軌道の係数  $c$  の決定について必要な式 (10) の並列化について述べる。式 (10) の一部を原子軌道基底で表記すると、

【数 13】

について各プロセッサで転送前に和を取ることで、 $N^2$  になり転送データ量が削減されている。

【0050】 最後に、エネルギー勾配計算に必要な、原子核座標による 2 電子積分の並列化について述べる。前述のように、本計算においては繰り返し計算の必要がないので、データ量が  $n^4$  と少ない分子軌道基底での  $\Gamma$  を各プロセッサに転送し、分子軌道基底から原子軌道基底に逆変換することで並列化する。その逆変換アルゴリズムを、図 4 に示す。ここでも、 $r$  と  $s$  で並列化し、 $t$ 、 $u$  の全てについて計算する構造は通常の 2 電子積分の場合と同様である。但し、D O ループの深さが 7 重になっている点が異なっている。各プロセッサで計算された式 (11) の第 2 項は、そのプロセッサ上で原子核に働く力を保存している配列に足し込まれた後ギャザリングされることになるが、その配列の大きさは  $3 \times$  原子数であり、転送ネックになることはない。また、 $\Gamma$  の逆変換に

対する演算数については、従来法と違いはない。

【0051】ここで述べた並列化アルゴリズムは、原子軌道のインデックス  $r$  と  $s$  で並列化するため、プロセッサ台数の増加に対して粒度を均一に保つことができ、高いスケーラビリティを維持することができる。また、適応可能なプロセッサ台数は  $N^2$  であり、 $N=1000$  で百万台の並列コンピュータに対応できることになる。

2 電子積分変換における従来変換方式と本方式との演算量と作業領域の比較

	N <sup>5</sup> 乗方式		本方式 (注1)	
	演算量	作業領域	演算量	作業領域
ステップ1	$N^4 n$	$N^4 + N^3 n$	$N^2 n$	$N^2 + N n$
ステップ2	$N^3 n^2$	$N^3 n + N^2 n^2$	$N n^2$	$N n + n^2$
ステップ3	$N^2 n^3$	$N^2 n^2 + N n^3$	$n^3$	$n^2 + n^3$
ステップ4	$N n^4$	$N n^3 + n^4$	$n^4$	$n^3 + n^4$
ギャザリング	—	—	$N^2 n^4$	

注1)  $N^2$  個のプロセッサを使用した場合

【表2】

分子軌道係数決定における従来変換方式と本方式との演算量と作業領域の比較

	N <sup>5</sup> 乗方式		本方式 (注1)	
	演算量	作業領域	演算量	作業領域
ステップ1	$N^4 n$	$N^4 + N^3 n$	$N^2 n$	$N^2 + N n$
ステップ2	$N^3 n^2$	$N^3 n + N^2 n^2$	$N n^2$	$N n + n^2$
ステップ3	$N^2 n^3$	$2(N^2 n^2)$	$N n^3$	$n^2 + N n^2$
ステップ4	$N n^4$	$2(N^2 n^3)$	$N^2 n^4$	$N n^3 + N n^3$
ギャザリング	—	—	$N^2$	

注1)  $N^2$  個のプロセッサを使用した場合

【0054】

【発明の効果】以上説明したように本発明によれば、複数のプロセッサを有するパラレルコンピュータを用いた計算方法であって、2電子積分の原子軌道基底 ( $r s | t u$ ) から分子軌道基底 ( $a b | c d$ ) への変換の際に、前記複数のプロセッサに対して、計算すべき原子軌道のインデックス  $r$  と  $s$  を指定し、前記複数のプロセッサの各々において、指定されたインデックス  $R$  と  $S$  と、計算すべき原子軌道のインデックス  $t$  と  $u$  の全ての組合せとについての処理を行う並列計算方法が得られ、これにより、粒度が均一になり、高性能下でコモディティプロセッサを多数接続することができ、高速演算コンピュータの費用低減が可能になり、更に、廉価なロカルメモリを活用できるので、コンピュータ全体として大きな主記憶領域を確保できる。

【図面の簡単な説明】

【0052】下記の表1及び表2に、従来の計算方法と本方法の演算数と作業領域の計算式を示した。並列化により演算数は若干増えているが、各プロセッサにおける作業領域は  $N^2$  分の1になっており、廉価なロカルメモリを有効利用できるアルゴリズムになっている。

【0053】

【表1】

	N <sup>5</sup> 乗方式		本方式 (注1)	
	演算量	作業領域	演算量	作業領域
ステップ1	$N^4 n$	$N^4 + N^3 n$	$N^2 n$	$N^2 + N n$
ステップ2	$N^3 n^2$	$N^3 n + N^2 n^2$	$N n^2$	$N n + n^2$
ステップ3	$N^2 n^3$	$N^2 n^2 + N n^3$	$n^3$	$n^2 + n^3$
ステップ4	$N n^4$	$N n^3 + n^4$	$n^4$	$n^3 + n^4$
ギャザリング	—	—	$N^2 n^4$	

注1)  $N^2$  個のプロセッサを使用した場合

【図1】本発明による2電子積分変換におけるプロセッサ間での分散処理の前半部分を示した図である。

【図2】本発明による2電子積分変換におけるプロセッサ間での分散処理の後半部分を示した図である。

【図3】本発明による分子軌道係数決定繰り返し計算におけるプロセッサ間での分散処理を示した図である。

【図4】本発明による2次の密度行列  $\Gamma$  の分子軌道基底から原子軌道基底への逆変換による並列化アルゴリズムを示した図である。

【図5】2電子積分の原子軌道基底から分子軌道基底への変換において現在使用されている  $4N^5$  変換アルゴリズムのステップ1～3を示した図である。

【図6】2電子積分の原子軌道基底から分子軌道基底への変換において現在使用されている  $4N^5$  変換アルゴリズムのステップ4を示した図である。

【図1】

分子軌道基底2電子積分生成並列化アルゴリズム

スレーブプロセス群

1) データ転送: 分子軌道の係数  $c$ 

→ [CPU#0]

→ [CPU#1]

→ [CPU#2]

3) 変換2

同左

同左

2) 変換1

R, S の指定

```

do t
  do u
    (RS|tu) の計算
  do d
    (RS|cd) = (RS|cd) +  $c_{du}$ (RS|tu)
  end do
end do
end do
do d
  do t
    do c
      (RS|cd) = (RS|cd) +  $c_{ct}$ (RS|tu)
    end do
  end do
end do
do c
  do d
    do B
      (RS|cd) = (RS|cd) +  $c_{cb}$ (RS|tu)
    end do
  end do
end do

```

ステップ1 →

ステップ2 →

ステップ3 →

同左

同左

4) ギャザリング:  $(ab|cd) = \sum_{t,u} (AB|cd)$ 

← [CPU#0]

← [CPU#1]

← [CPU#2]

【図4】

Γ逆変換によるエネルギー勾配計算並列化アルゴリズム

```

do r
  do s
    [r, s で並列化]
    do a
      do b
        do c
          do u
            do d
              Γ(abcu) ← Γ(abcd)
            end do
          end do
        end do
      end do
    end do
    do u
      do t
        do c
          Γ(abtu) ← Γ(abcu)
        end do
      end do
    end do
    do t
      do u
        Γ(aStu) ← Γ(abtu)
      end do
    end do
    do t
      do u
        Γ(RStu) ← Γ(aStu)
      end do
    end do
  end do
end do

```

2 電子積分  $(RS|tu)$  について,  $t, u$  全てについて計算し,  $\Gamma$  と掛ける

【図3】

分子軌道係数  $c$  の決定における並列化マスター  
プロセス

スレーブプロセス群

1) データ転送: 分子軌道係数  $c$ 

→ [CPU#0]

→ [CPU#1]

→ [CPU#2]

2) 変換1

図1のステップ1、2により  
( $c_{ab}|xy$ ) と  
( $c_{cd}|by$ ) を作成

同左

同左

3) データ転送:  $\Gamma$ 

→ [CPU#0]

→ [CPU#1]

→ [CPU#2]

4)  $V_{c,bd}(RS)$  の計算 $V_{c,bd}(RS)$  の計算

同左

同左

5) データ転送:  $u$ 

→ [CPU#0]

→ [CPU#1]

→ [CPU#2]

6)  $Y_{abcd}$  の計算 $Y_{abcd}$  の計算

同左

同左

7) ギャザリング:  $y = \sum_{t,u} Y$ 

← [CPU#0]

← [CPU#1]

← [CPU#2]

if u=y  
yes →  
収斂

【図 5】

従来の 2 電子積分 4N<sup>2</sup> 乗変換アルゴリズム

```

ステップ 1  do r
              do s
                do t
                  do u
                    do d
                      (rs|td) = (rs|td) + cud × (rs|tu)
                    end do
                  end do
                end do
              end do
            end do

ステップ 2  do d
              do r
                do s
                  do t
                    do c
                      (rs|cd) = (rs|cd) + ctc × (rs|td)
                    end do
                  end do
                end do
              end do
            end do

ステップ 3  do c
              do d
                do r
                  do s
                    do b
                      (rb|cd) = (rb|cd) + csb × (rs|cd)
                    end do
                  end do
                end do
              end do
            end do

```

【図 6】

```

ステップ 4  do b
              do c
                do d
                  do r
                    do s
                      (ab|cd) = (ab|cd) + cra × (rb|cd)
                    end do
                  end do
                end do
              end do
            end do

```

---

フロントページの続き

(71)出願人 394017491  
株式会社エヌイーシー情報システムズ  
神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号

(72)発明者 中田 一人  
東京都江東区新木場一丁目18番 6 号 エヌ  
イーシーソフト株式会社内

(72)発明者 村瀬 匡  
東京都江東区新木場一丁目18番 6 号 エヌ  
イーシーソフト株式会社内

(72)発明者 佐久間 俊広  
神奈川県川崎市高津区坂戸三丁目 2 番 1 号  
株式会社エヌイーシー情報システムズ内

(72)発明者 高田 俊和  
東京都港区芝五丁目 7 番 1 号 日本電気株  
式会社内

F ターム(参考) 4H006 AA05 AC90  
5B045 GG02 GC11  
5B056 AA04 BB04 FF05